
This is the **published version** of the bachelor thesis:

Ropero Mendez, Isaac; Alsina Rodriguez, Aitor, dir. Desenvolupament videojoc Unreal Engine 4 amb interacció a la partida des d'aplicació web. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/248501>

under the terms of the  license

Desenvolupament videojoc Unreal Engine 4 amb interacció a la partida des d'aplicació web

Isaac Ropero Méndez

Resum– A començaments d'una nova generació de videoconsoles, i al davant d'un canvi generacional en el qual hi ha un factor important, com és el fet que persones amb accés a un públic molt ampli busquen una interacció directa i dinàmica amb l'audiència, neix la idea d'oferir noves jugabilitats que anteriorment no es consideraven. El treball final de grau consisteix en desenvolupar un videojoc utilitzant el motor gràfic Unreal Engine 4 de Epic Games, amb interacció amb altres jugadors que fan servir un client web de navegador. L'intercanvi de missatges es realitza a través d'un servidor NodeJS que utilitza la tecnologia Websockets pel control del flux de la comunicació.

Paraules clau– Unreal Engine 4, UE4, NodeJS, VUE, Client web, HTTP, Websocket, C++, HTML, Javascript, Videojoc, Top-down games

Abstract– At the beginning of a new generation of video game consoles, and in the face of a generational change in which we have an important factor such as the fact that people who have access to a very wide audience looking for direct interaction with the audience. The dissertation involves developing a video game using Epic Games' Unreal Engine 4 graphics engine, interacting with other players using a web browser client. The exchange of messages is done through a NodeJS server that uses Websockets technology to control the flow of communication.

Keywords– Unreal Engine 4, UE4, NodeJS, VUE, WebClient, HTTP, Websocket, C++, HTML, Javascript, Game, Top-down games



1 INTRODUCCIÓ

EL desenvolupament d'un videojoc és una tasca que implica un gran esforç, tenir coneixements de diferents àrees, disposar d'un equip qualificat i invertir una quantitat de temps considerable per aconseguir un producte de qualitat. Fins i tot, els nous tipus de videojocs que són populars actualment, els quals són els desenvolupats per poques persones o inclús per un únic membre, han de centrar molts esforços a crear mons virtuals on la part visual sigui menys important, però que siguin capaços de submergir al jugador en una experiència única i gratificant.

S'ha convertit quasi en tradició, que la majoria de jocs comercials hagin de comportar-se com els de tota la vida, és a dir, nosaltres com a jugador superant la història. En-

cara que canviï l'estil del joc sempre som nosaltres els que decidim com volem influenciar en aquesta partida. Cada cop els usuaris demanen més control per a poder prendre decisions, que aquestes afectin el món, i que no generin la sensació que la decisió no era rellevant, creant al final una jugabilitat on no importa fer una cosa o altra. És en aquest context on entra en escena el nostre videojoc. Encara que no es basa a crear una jugabilitat predissenyada que sigui molt immersiva, sinó a oferir la possibilitat de crear la jugabilitat en temps real, a mesura que els esdeveniments van succeint.

En aquest article s'explica l'evolució al llarg del procés de desenvolupament, el qual el formen plantejar la idea, conèixer les tecnologies disponibles, les eines que existeixen, la comunicació entre els components, el disseny del videojoc, el disseny de personatges, les textures, la il·luminació i les animacions, fins a poder mostrar el resultat final en una petita demo.

L'estructura que segueix el document és en primer lloc descriure l'escenari actual en el sector dels videojocs sobre tecnologies existents, tot seguit de definir els objectius del projecte. A continuació la metodologia escollida i la planificació que s'ha realitzat. La resta d'apartats expliquen el

- E-mail de contacte: isaac.ropero@e-campus.uab.cat
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Aitor Alsina Rodriguez (departament)
- Curs 2020/21

disseny i la implementació del videojoc, i l'apartat de resultats mostra les diferents proves que s'han realitzat per demostrar el funcionament.

2 ESTAT DE L'ART

Actualment les eines disponibles per realitzar connexions entre un joc en Unreal Engine 4 i un servidor se solen veure en forma de plug-in desenvolupats per la comunitat. Hi ha alguns com VaRest [1] o similars [2] que ofereix la possibilitat de programar aquestes funcionalitats, com poden ser instruccions per realitzar peticions a un servidor, utilitzant el que es coneix com a blueprints [3]. Blueprint és un sistema que incorpora Unreal que permet programar funcionalitats de forma gràfica. VaRest, i com molts altres, són de pagament. Normalment aquests projectes solen oferir també el repositori de Git per descarregar el codi font i compilar el plug-in, tot i que no solen ser l'última versió compatible amb Unreal Engine.

Fer servir aquestes eines té com a inconvenient, que a part de ser de pagament i que no totes ofereixen el codi, el desenvolupador està limitat a fer servir les funcionalitats del plug-in, quan potser s'han d'adaptar al projecte, com podria ser per exemple, el format de l'intercanvi de missatges.

Per altra banda, el mateix motor de Unreal Engine disposa de llibreries per controlar connexions cap a serveis externs, com poden ser HTTP o WebSocket [4]. És responsabilitat del desenvolupador documentar-se i comprendre com funcionen aquestes llibreries, i ajudar-se d'exemples realitzats per la comunitat per implementar aquest disseny a les funcionalitats requerides. Com a part negativa és que per la programació ja no es pot realitzar utilitzant Blueprints i s'ha d'implementar en C++, i la documentació oficial per part de Unreal Engine no és gaire bona.

3 OBJECTIUS

El propòsit d'aquest projecte és el de permetre a un conjunt de jugadors poder crear una partida i jugar entre ells. On un dels jugadors, serà el director del joc, i el seu objectiu principal serà el de controlar la partida i oferir reptes a la resta de jugadors. Això ho podrà fer a través d'un portal web dinàmic que li permetrà interactuar amb el joc, podrà crear, prendre decisions, modificar paràmetres de la partida en temps real, tot això i altres per oferir als jugadors interaccions amb les partides per tal que siguin dinàmiques. La resta de jugadors, únicament es connectaran a la sessió de joc que el director els hi haurà preparat, i jugaran la partida, mesclant accions que el joc té programat de forma estàtica, com pot ser recórrer un escenari i lluitar contra algun enemic, i altres situacions on hauran de resoldre els reptes que el director els hi hagi preparat, com per exemple, poder escollir diferents camins per continuar i haver de triar un o altre, i com tot això ho ha preparat el director, només ell sap el que passarà.

Per categoritzar els objectius s'han dividit en dues seccions, objectiu principal i objectius específics. L'objectiu principal és el que s'ha d'assolir per finalitzar el projecte i els objectius específics són aquests que aporten valor i millores, i en cas que algun no es pugui completar no afecta la base de la jugabilitat.

3.1 Objectiu principal

- Implantació i programació del servidor web que s'en-carregui de controlar, protegir i comunicar els clients.
- Desenvolupament del client de navegador, el qual ha de ser dinàmic i permetre escalabilitat.
- Desenvolupament del videojoc, i tot el que és es considera mínimament necessari per al funcionament d'aquest, com els objectes 3D, escenari principal, i comunicació, entre altres.
- Permetre a l'usuari la creació de sessions privades per a cada partida protegides amb contrasenya.
- Permetre a l'usuari crear i unir-se a les partides sense requerir un software específic.

3.2 Objectius específics

- Implementar diferents modes de jugabilitat, com escollir quin personatge triem per cada partida i les diferents habilitats així com escenaris diferents.
- Creació d'un xat per interactuar entre el jugador principal i els secundaris dins del joc.
- Poder guardar en disc, tant la partida, com les sessions i dades dels jugadors.
- Millorar l'aspecte gràfic i immersió del videojoc, així com millorar la jugabilitat i permetre partides més llargues i complexes.

El problema que planteja aquest tipus de projecte és quin software s'escollirà per fer el videojoc així com el llenguatge de programació. Un apartat molt important és com es realitzarà la connexió entre client web i client del videojoc amb el servidor, i quina tecnologia o protocol pot ser més útil en aquest escenari, així el com les dades seran administrades, en especial, les de l'usuari que accedirà al portal web, ja que pot ser necessari implementar una base de dades. La seguretat és un punt de molta importància, ja sigui pel que fa a la confidencialitat de les connexions, és a dir que siguin segures i privades, tant com per garantir tot el relacionat amb no permetre les trampes, ja que un jugador podria arribar a comprometre el sistema per obtenir avantatges.

4 METODOLOGIA

La metodologia utilitzada per al desenvolupament d'aquest projecte es basa en un model de prototips. És un model en el qual el prototip és construït, provat i reconstruït de nou fins que s'aconsegueix una versió satisfactòria. El prototip pot ser avaluat en cada moment del desenvolupament. Les etapes són les següents:

1. Anàlisi de requeriments.
2. Disseny simple.
3. Realització del prototip.
4. Avaluació de l'usuari.

El desenvolupament d'aquest projecte s'ha basat en la implementació de cinc prototips:

1. Prototip que forma en el nivell de proves inicials.
2. Prototip amb el menú, personatge i canvi d'escenari.
3. Prototip amb el menú, connexió amb el servidor, personatge i canvi d'escenari.
4. Prototip amb el menú que permet escollir servidor, sessió i contrasenya, connexió amb el servidor, personatge i canvi d'escenari.
5. Prototip amb el menú que permet escollir servidor, sessió i contrasenya, connexió amb el servidor, canvi d'escenari, es pot escollir personatge, intercanvi de missatges per interactuar entre el navegador i el videojoc i nivell principal acabat.

El primer prototip consisteix en el nivell de proves, el qual permetrà al principi tenir una base on començar a provar els diferents objectius que es volen aconseguir. Les següents iteracions consisteixen a introduir funcionalitats noves, com el menú, el canvi de nivell, el personatge i el prototip més actual, el qual ja inclou els diferents submenús dins del menú principal, el primer nivell de la masmorra, algunes opcions que el director de joc pot configurar a l'inici de la partida i alguns esdeveniments per demostrar futures funcionalitats.

Dins d'aquesta metodologia, s'han seguit diferents línies, la dels prototips per part de client web i les del client del videojoc.

1. Prototip creació servidor web i component inicial dinàmic.
2. Prototip gestió de la comunicació des de client web cap a servidor
3. Prototip gestió de la comunicació bidireccional

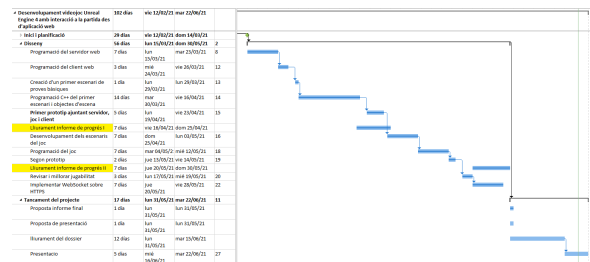
5 PLANIFICACIÓ

La data límit per la realització del treball té una estimació d'unes 300 h des del començament del projecte. Per dur a terme el projecte s'ha realitzat una estimació amb el MS Projects, així com la realització d'un diagrama de Gantt.

La planificació està formada per diferents fases:

- Inici i planificació
- Disseny
- Tancament del projecte

A la figura (1) es pot veure el diagrama de Gantt que s'ha seguit en la planificació del projecte, on es pot observar les fases mencionades i la seva durada prevista.



un compte registrat en la seva plataforma, aleshores després d'un any de contingut gratuït el qual pot ser totalment utilitzat tant per productes comercials com privats, la quantitat de assets que podien ser útils per aquest projecte era un punt molt a tenir en consideració. Per altra banda, existeix l'estereotip del fet que Unreal costa molts diners, que la dificultat d'aprenentatge és molt elevada, que només serveix per desenvolupar jocs d'estil de trets i sembla que Unity hagi de ser la solució a tot això, així que un altre motiu, potser de menys importància, va ser donar una oportunitat a Unreal.

8 IMPLEMENTACIÓ

La implementació es divideix en les següents seccions i detallen la seva definició.

8.1 Videojoc UE4

Com la secció d'implementació del videojoc és molt extensa, ja que té molts components diferents, en aquesta part només s'analitza la part de comunicació amb els altres dos mòduls del projecte. Tota la part del disseny del videojoc i dels seus components és explicat en l'apartat de resultats.

8.1.1 Comunicació en xarxa

Per poder realitzar la comunicació entre el client i el servidor s'han fet servir els següents mòduls de Unreal Engine:

- Http
- Json
- JsonUtilities
- WebSockets

El mòdul de HTTP permet controlar tot el que està relacionat amb el protocol http, com per exemple fer un request, o obtenir o definir les capçaleres dels missatges, entre moltes altres funcionalitats.

El mòdul de JSON i JSONUtilities permet disposar de diferents mètodes ja definits per l'administració d'objectes JSON. Com per exemple, convertir un JSON a una estructura de dades. En unreal treballar amb JSON pot arribar a ser bastant complex, sobretot amb objectes que contenen altres objectes, ja que el programador ha de definir l'estructura en funció a la cadena JSON que rebrà.

El mòdul de Websockets permet crear enllaços dinàmics entre les funcions del protocol websocket i funcions que nosaltres definim, com per exemple, el missatge de websocket *on()*, podem fer que quan aquest succeeix es cridi a una funció que nosaltres definim.

La comunicació s'estableix quan l'usuari des del menú principal del videojoc prem el botó de connectar-se. En aquest moment internament, el primer que fa el joc és connectar-se per HTTP al servidor Node, si aquest es troba operatiu. Un cop s'ha connectat i ha rebut una resposta vàlida per part del servidor, estableix una connexió Websocket. A partir d'aquest moment intercanvien servidor i

client un missatge de confirmació utilitzant el canal establert i pot començar la partida.

Una de les diferències en complexitat a l'hora de programar el client del joc en comparació amb el client de navegador, és que el navegador moltes coses les realitza de forma implícita, cosa que en UE4 no, per exemple en unreal totes les capçaleres necessàries s'han de definir i s'han de crear estructures de dades per tractar la informació per posteriorment poder enviar i rebre dades. Així com tenir en consideració el funcionament dels protocols HTTP, Websocket i el tipus de missatges que intercanvien.

8.2 Servidor

Per la part de programació del servidor s'ha utilitzat NodeJS, juntament amb paquets addicionals com *session*, que permet crear un middleware pel control de les sessions, amb aquest mòdul en les cookies només enviem el ID de la sessió, i *express*, el qual ens permet definir una aplicació web que ja proporciona moltes funcionalitats implementades i fàcils d'utilitzar. NodeJS va ser escollit per la simplicitat que es necessita per posar en marxa el servidor des de la línia d'ordres. Per la comunicació s'ha decidit fer servir la tecnologia *Websocket* que és una tecnologia que permet l'intercanvi de missatges per un canal de comunicació bidireccional i full-dúplex fent servir un únic socket TCP.

Fins al moment, la funcionalitat del servidor és la següent:

- Utilitzar sessions per poder validar a l'usuari un cop es connecti al servidor HTTP.
- Validar usuaris fent servir un inici de sessió a través de HTTP
- Diferenciar entre usuari client web (director de joc) i usuaris de unreal engine
- Crear partides, és a dir, definir els paràmetres de connexió pel client unreal engine.
- Administrar i permetre connexions Websocket
- Control d'una estructura de dades interna que de moment compleix la funció de base de dades dels usuaris.
- Permet la interacció entre el videojoc i els clients web.

Per la part d'informació enviada entre client i servidor, pot ser que tant com cadenes de text, com estructures Json.

8.2.1 Seguretat

Un dels aspectes de seguretat és el fet d'utilitzar contrasenyes per protegir les sessions que crea el director del joc per evitar l'entrada a la partida de tercers jugadors no desitjats. Això, tot i afegir una capa de seguretat, la comunicació entre client i servidor no és xifrada, així que aquestes contrasenyes poden ser capturades. Un dels últims apartats que es vol dur a terme durant el desenvolupament del projecte és implementar HTTPS i SSL sobre websocket.

Fins al moment hi ha alguns aspectes de seguretat que s'han contemplat, com pot ser que un usuari de navegador no pugui enviar ordres que només el client de Unreal pot fer, o per altra banda que enviés ordres que per mecàniques internes del joc no ha de poder fer si no compleix algun requisit. Hi ha certes ordres o funcions que només pot fer un client web o per altra banda el joc, i aquestes són restrictives al tipus d'usuari.

Un dels objectius que falta implementar cara la pròxima entrega són comunicacions HTTPS.

8.3 Client web

El client web està desenvolupat en HTML juntament amb VUE i Bootstrap, això permet tenir una pàgina web adaptativa i dinàmica. La finalitat del client web és oferir una interfície que mostri fidelitat al disseny i atmosfera gràfica que acompanya al joc, i que al mateix temps les diferents seccions que formen aquesta UI estiguin separades en mòduls, particularment en components VUE. Aquests mòduls poden no canviar al llarg de tota la partida, o depèn la secció en la qual es troba el jugador principal poden ser diferents.

9 RESULTATS

9.1 Videojoc UE4

Per la part del videojoc amb Unreal Engine es divideix en diverses seccions, tant com pot ser la comunicació en xarxa, el disseny, el modelatge o personatges, les animacions, l'àudio, la història, la jugabilitat, entre d'altres.

En l'apartat A.2 podem trobar el diagrama de casos d'ús que mostra les accions que el joc pot fer.

9.1.1 Disseny del joc

El joc es basa en una modalitat top-down, on la càmera es troba just a sobre del jugador, orientada cap a baix, i la vista que ofereix és com si estiguéssim sobre el jugador mirant cap a sota.

La jugabilitat consisteix en un jugador amb el rol de director de la partida el qual s'encarregarà de crear els personatges d'aquella sessió juntament amb els paràmetres d'aquesta, com pot ser el nom de la sessió i la contrasenya. En aquest projecte, només es crearà un únic personatge que podrà ser controlat pel jugador, el qual serà creat pel director, com si es tractés d'omplir una fulla d'un joc de rol de taula.

Un cop la partida està creada, els jugadors de Unreal Engine intentaran superar el nivell, el qual ja està predissenyat pel desenvolupador, el director de joc el que pot fer és crear esdeveniments o modificar algunes parts de l'escenari perquè succeeixin esdeveniments que ell decideix per ajustar la partida, afegint o traient dificultat, o creant situacions diferents en cada partida. Al final, el rol del director no deixa de ser tenir una visió global del que està passant i poder decantar les situacions cap a un lloc o altre. I els jugadors senzillament intentaran escapar de l'escenari, col·laborant o no entre ells.

9.1.2 Assets

Pel que fa a tot el material gràfic, tant com personatges, animacions, textures, àudio, entre altres, s'han obtingut del contingut gratuït que distribueix Epic Games en la seva botiga virtual, tant com el material que regala de forma mensual i s'ha anat acumulant al llarg d'un any aproximadament.

Hi ha una part dels assets que han sigut obtinguts d'altres fonts, o han sigut modificats amb programes d'edició, com alguns fitxers d'àudio.

El resta de contingut, i especialment animacions i models 3D de personatges, figura (2) i (3), s'han obtingut de Mixamo, per part de Adobe.

Alguns objectes sí que s'han realitzat per part pròpia. Com poden ser algunes textures que s'han creat a partir de fotografies reals i aplicant normal maps.

9.1.3 Personatges

Durant la creació de la partida el director pot escollir el personatge per cada jugador. En el projecte s'han utilitzat dos personatges diferents. Un que realitza el rol de lluitador, i un altre que realitza el rol de druida.

Lluitador El lluitador és el personatge que pot equipar-se amb armes cos a cos, té més quantitat de salut, i el seu punt fort és la força per fer front als enemics.



Fig. 2: Personatge jugable del joc.

Druida El druida és el personatge que realitza la funció de suport, el seu objectiu és mantenir amb vida la resta de jugadors fent servir màgies de recuperació de vida. Al mateix temps també domina l'habilitat de convertir-se en diferents animals, com podria ser un tigre o un ós.

9.1.4 Escenari de proves i testatge de funcionalitats bàsiques

Per comprovar els avanços realitzats i analitzar que funciona correctament tant per la part del servidor com per la part del joc s'ha creat un escenari de proves bàsic.

Aquest escenari és molt bàsic i només es va fer servir per provar les entrades de teclat i ratolí, així com les animacions de moviment, la connexió amb el servidor, i alguns esdeveniments entre client i servidor, com enviar missatges, generar un acte en pressionar certs objectes, o crear un actor en el nivell des del navegador. Figura (4).



Fig. 3: Personatge jugable del joc.



Fig. 4: Escenari de proves.

9.1.5 Menu principal

En unreal engine per crear interfícies d'usuari s'utilitzen objectes de tipus *Widget Blueprint*. Aquests ens permeten arrossegar components gràfics com botons o textos sobre un panell el qual es pot vincular a la vista de l'escena d'un nivell. De tal forma que l'usuari pot interactuar amb aquest panell.

El menú principal ens permet unir-nos a una partida o tancar el joc, figura (5). Si escollim unir-nos a una partida ens apareix un submenú en el qual s'ha d'indicar a quin servidor en volem connectar, quin port, quina sessió i quina contrasenya. Si la connexió s'ha pogut realitzar correctament, el botó de començar s'habilitarà, figura (6). Per mostrar les diferents opcions així com el fons s'ha escollit una temàtica tipus RPG.



Fig. 5: Menú principal.

9.1.6 Inici del primer escenari

El primer escenari és més ambiciós i ja contempla elements artístics, així com il·luminació, interacció amb el nivell, partícules, entre altres, tot i que no és l'escenari final, sinó una part inicial del conjunt del nivell. Figura (7) i (8).

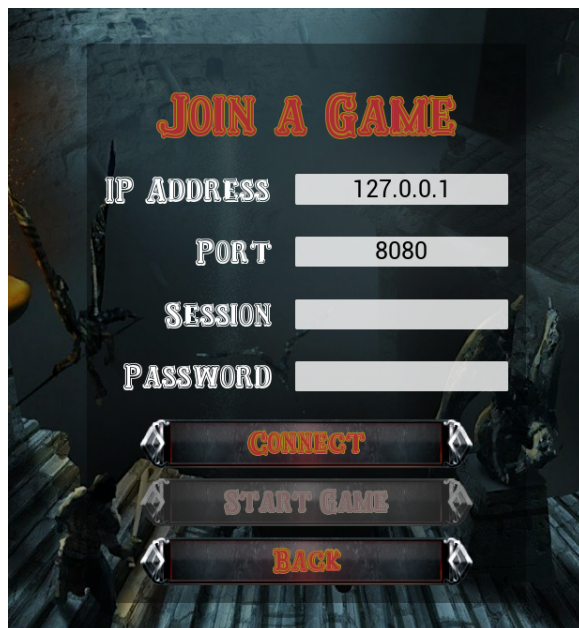


Fig. 6: Menú de conexio al servidor.

En l'editor de Unreal hi ha la possibilitat de generar terrenys o conjunts d'objectes de tal forma que es puguin dibuixar directament sobre l'escenari. Durant la creació de l'escena alguns dels assets s'han distribuït així, altres, s'ha pensat en detall com distribuir-los sent fidel al disseny artístic del videojoc.



Fig. 7: Part d'una escena del nivell de la masmorra.



Fig. 8: Escena del pont del nivell.

9.1.7 Interfície d'usuari

La interfície d'usuari en aquest joc és molt senzilla, torna a ser un altre Widget que només mostra dos *progress bars*. La interfície està vinculada als atributs de la classe que controla els personatges, així quan el valor és modificat en l'àmbit de codi aquest es veu reflectit en la pantalla indiferentment de quin personatge s'estigui controlant. Figura (10).



Fig. 9: Barra de salut i poder màgic.

9.1.8 Il·luminació i àudio

La part d'il·luminació d'una escena és complexa i molt extensa com per descriure tot en aquest article, indicar que hi existeixen diferents fonts de llum, com poder ser point lights, la qual es mostra com si tinguéssim una bombeta per l'escena que il·lumina amb una intensitat i una distància que indiquem, després tenim altres fonts de llum, com la spot lights, que el seu comportament és com d'un focus, el qual podem indicar el radi d'obertura intern i extern i la distància de la llum, i finalment tenim llums com la direccional, que simulen el comportament del sol.

Per la part de l'àudio, també hi ha diferents tipus, però en resum en aquest projecte s'han fet servir sons ambientals, com pot ser el que s'escolta contínuament de fons durant la partida. Aquest tipus és més especial, perquè són cues de molts àudios diferents que de forma aleatòria van canviant, això crea una sensació d'immersió i no genera l'efecte d'estar sentint tota l'estona un bucle.

9.1.9 Animacions de moviment

Per la part d'animació, que és un aspecte que també repercuteix en molta feina, s'ha optat en fer servir totes les animacions possibles que Mixamo ens pot oferir de forma gratuïta. Dit això, Unreal disposa d'una eina que ens permet definir un objecte de tipus blendspace, que aquest ens permet fer transicions entre diferents animacions perquè siguin més fluides i naturals, com pot ser per exemple, la transició entre estar quiet, caminar o córrer. Figura (11).

Un cop s'ha creat i configurat el blendspace, es passa a crear un altre objecte diferent que es tracte particularment d'un blueprint d'animacions. Aquest ens permet definir un diagrama d'estats que proporciona el control entre quan volem que es compleixi una condició perquè aquestes transicions puguin començar, per exemple, obtenim la velocitat actual de l'actor al qual assignem aquest blueprint, i segons la velocitat de moviment, el diagrama canviarà d'estat i complirà les transicions indicades en el blendspace.

9.2 Client Web

El client web, següidament de fer la validació i connexió amb el servidor, ens permet definir les característiques de la partida en el full de dades del jugador com es pot veure a la figura (15).



Fig. 10: Objectes d'àudio i de il·luminació.

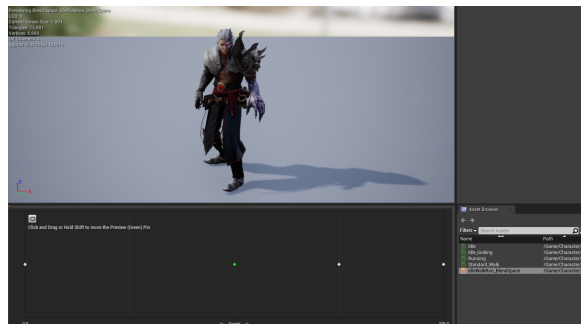


Fig. 11: Objecte blendspace, transició entre estar quiet, caminar o córrer.

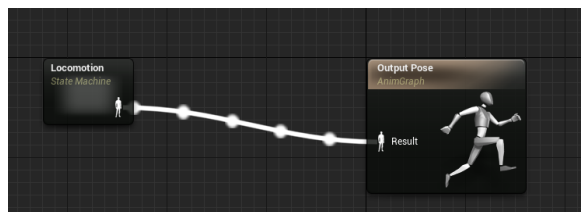


Fig. 12: Diagrama d'estats del blueprint d'animacions.

Un cop la partida està creada, es pot interactuar amb el videojoc a través dels components de vue, que són mòduls que el director de joc fa servir per enviar ordres. Els mòduls disponibles es poden veure en la figura (14).

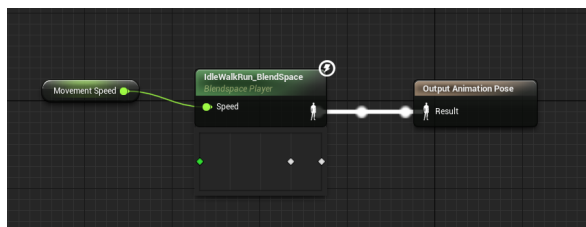


Fig. 13: Interactuar amb un blendspace dins d'un diagrama d'estats d'animació.

Fig. 14: Opcions disponibles pel director de joc.

Fig. 15: Full de dades del jugador.

10 CASOS D'US

En la secció A.4 es mostra l'esdeveniment que succeeix quan el jugador se situa just al damunt del quadrat. Canviant el color del component VUE de vermell a verd.

En la secció A.5 es mostra l'esdeveniment que succeeix quan des del navegador web, un usuari prem el botó de *Spawn Enemy*, això provoca que en el joc aparegui un llop, que és un enemic del jugador principal. Actualment encara no hi ha intel·ligència artificial.

En la secció A.6 es mostra l'esdeveniment que succeeix quan des del navegador web, un usuari prem el botó de *Spawn Item*, això provoca que en el joc aparegui l'objecte seleccionat, que pot ser o una poció o una bomba. Tots dos objectes són funcionals, si el jugador agafa la poció recuperarà 20 punts de salut i si agafa la bomba aquesta explotarà i li prendrà vida.

En la secció A.7 es mostra l'esdeveniment que succeeix quan des del navegador web, un usuari activa la plataforma, aquesta dins del joc començarà a moure's completant un recorregut, d'aquesta forma el jugador podrà creuar el pont.

11 CONCLUSIONS

L'objectiu al qual es volia arribar en el projecte era el de poder crear un videojoc amb la plataforma de Epic Games que aquest fos capaç de comunicar-se amb una aplicació de navegador, mitjançant un servidor web que seria l'encarregat de gestionar les connexions, sessions i l'intercanvi de missatges entre aquests dos. També es volien aconseguir objectius més ambiciosos, com un cop obtinguda la funcionalitat bàsica, fer el mateix però utilitzant un canal segur xifrat, i després, referent a la part creativa, es volia aconseguir oferir a l'usuari, tant el que realitza el rol de director com al jugador, un món interactiu que fos atractiu i oferís als jugadors una experiència agradable i que es quedés amb ganes de tornar.

Al finalitzar el treball, s'ha aconseguit realitzar la comunicació i que aquesta sigui eficient i segura, tot i que la part del xifratge no està implementada la part que comprova que els jugadors no puguin fer trampes està finalitzada. Per altra banda s'ha dissenyat un nivell inicial suficientment atractiu per mostrar al jugador tot el potencial que aquest joc podria oferir, el nivell està format per tots els objectes i funcionalitats que tindria un escenari en un joc comercial actualment, i realitzat amb programació d'alt nivell com C++ en compiles de Blueprints.

Aquest projecte ofereix com a avantatge la possibilitat de desvincular el videojoc d'algun servei ja existent i que no siguin de pagament. Tot i que és necessari els coneixements per realitzar les comunicacions, un cop això s'aconsegueix, l'avantatge és que podem fer servir qualsevol plataforma, inclús realitzada pel mateix desenvolupador, i administrar tot el control d'intercanvi de missatges. També ofereix la possibilitat de no dependre d'una tecnologia, ja que en aquest projecte s'ha utilitzat navegador web, però amb aquest mateix mecanisme s'hagués pogut implementar per telèfons intel·ligents o tauletes, i per diferents sistemes operatius.

Pel que fa a la part inicial de decidir quin motor gràfic s'utilitzarà en un videojoc, després de completar el projecte s'arriba a la conclusió que depèn més aviat de factors externs, com per exemple, si publiquem el joc, en quina plataforma de venda es farà, o el tipus de llicència del projecte, a quin nombre de còpies venudes creiem que arribarà i aquests tipus d'indicadors. Per la part tècnica, existeix un mite al voltant de unreal engine de què és molt difícil i cal començar sempre per unity. Aquest pensament és una idea equivocada, unreal al principi pot ser difícil, però perquè el motor té una quantitat enorme de mòduls que permet realitzar moltes funcionalitats, i la quantitat d'informació que hi ha al començament és molta, però això no vol dir necessàriament complexitat, sinó que l'eina ofereix una suite de productes que satisfan les necessitats d'un desenvolupador de videojocs.

Com experiència personal obtinguda al llarg del procés d'aprenentatge i desenvolupament, és inevitable sentir-se aclaparat per la quantitat de diferents àrees de treball que existeixen en la realització d'un videojoc i la quantitat de

gent que es pot arribar a necessitar per realitzar jocs professionals com els que podem trobar en el mercat. En aquest projecte, s'ha tingut la necessitat d'encarregar-se de tots, i com a part positiva això permet obtenir coneixements de moltes àrees diferents sacrificant en part la qualitat i quantitat dels resultats finals, però deixa una porta oberta per poder seguir millorant aquest projecte en el futur, o començar un altre, després d'obtenir una bona base tècnica.

11.1 Treball futur

Donat que el temps disponible per dur a terme el projecte és limitat i per les dificultats que han pogut anar sorgint al llarg del desenvolupament no s'han pogut dur a terme tots els objectius inicialment plantejats. Tot seguit es descriuen les millores que s'haurien d'implementar en futures línies de treball.

- Permetre l'intercanvi de missatges fent servir HTTPS, i protegir les dades perquè no viatgin en clar.
- Implementar un sistema d'invitació als jugadors a part de l'existent que permeti convidar a la partida a través de la plataforma on el joc estigui publicat, com podria ser Steam o Epic Games.
- En el cas de seguir en la línia de client web, considerar la idea de fer servir algunes de les eines de Unreal Engine per a navegador, com podria ser la creació d'un videojoc directament per navegador com a plataforma fent servir HTML5 i el mateix motor gràfic ofereix els recursos.
- Si el servidor no ha de controlar moltes peticions, es pot considerar migrar l'entorn a PHP.
- Continuar el disseny actual del nivell i dissenyar nous assets per la resta de nivells per acabar el videojoc.
- Programació d'un menú de pausa que permeti sortir del joc o modificar configuracions com àudio i vídeo

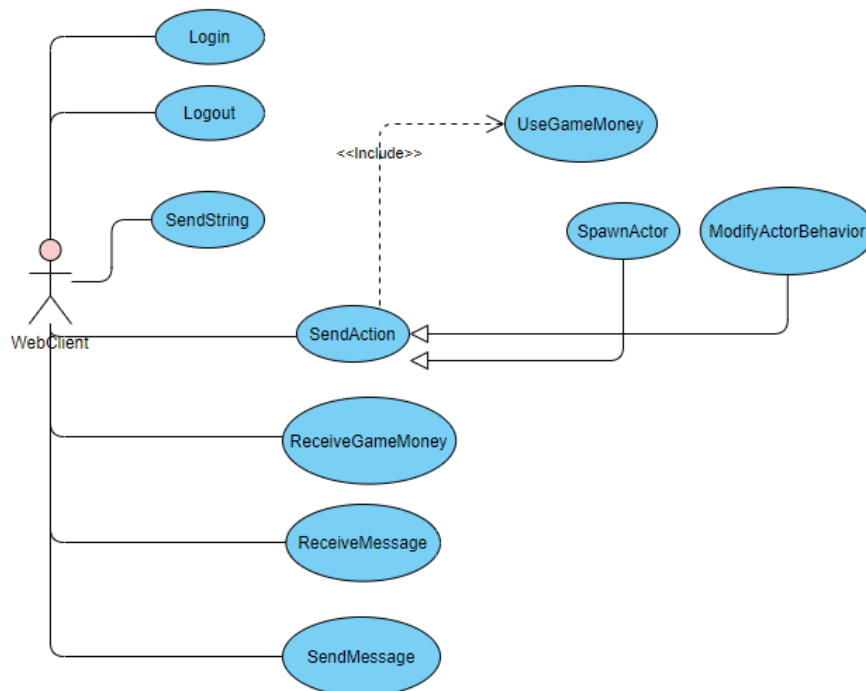
- [9] <https://forums.unrealengine.com/community/general-discussion/1587062-how-much-is-learning-curve-for-unreal-engineE>
- [10] <https://www.mixamo.com/>
- [11] <https://docs.unrealengine.com/en-US/ProgrammingAndScripting/ProgrammingWithCPP/index.html>
- [12] <https://www.gamedev.tv/p/unrealmultiplayer>
- [13] <https://ue4community.wiki/legacy/http-requests-hjqds788>
- [14] <https://ue4community.wiki/legacy/ue4.10-how-to-make-http-get-request-in-cpp-ektwg52q>
- [15] <https://forums.unrealengine.com/development-discussion/c-gameplay-programming/102732-game-as-socket-server>
- [16] <https://luisjordan.net/node-js/node-js-vs-php-comparaciones-cms-beneficios-y-tiempos-de-respuesta/>
- [17] <https://nodejs.org/en/>
- [18] <https://vuejs.org/>
- [19] <https://ue4community.wiki/websocket-client-cpp-5vk7hp9e>
- [20] <https://www.npmjs.com/package/express-session>
- [21] <https://expressjs.com/es/>
- [22] <https://github.com/websockets/ws/blob/master/examples/express-session-parse/index.js>

REFERÈNCIES

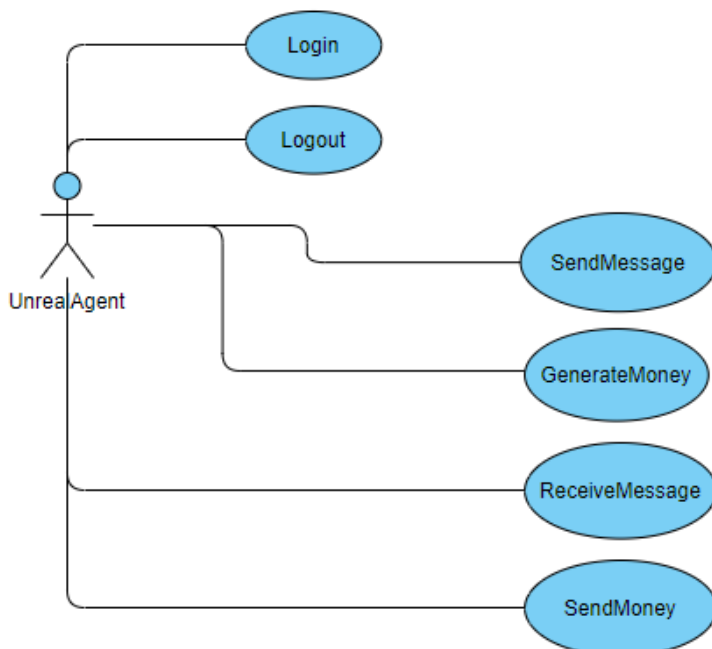
- [1] <https://www.unrealengine.com/marketplace/varest-plugin>
- [2] <https://github.com/getnamo/socketio-client-ue4>
- [3] <https://docs.unrealengine.com/en-US/ProgrammingAndScripting/Blueprints/index.html>
- [4] <https://es.wikipedia.org/wiki/WebSocket>
- [5] <https://en.wikipedia.org/wiki/Ninth-generation-of-video-game-consoles>
- [6] <https://www.unrealengine.com/en-US/blog/a-first-look-at-unreal-engine-5>
- [7] <https://es.wikipedia.org/wiki/Videojuego-de-disparos-en-primera-persona>
- [8] <https://docs.unrealengine.com/en-US/ProgrammingAndScripting/ProgrammingWithCPP/index.html>

APÈNDIX

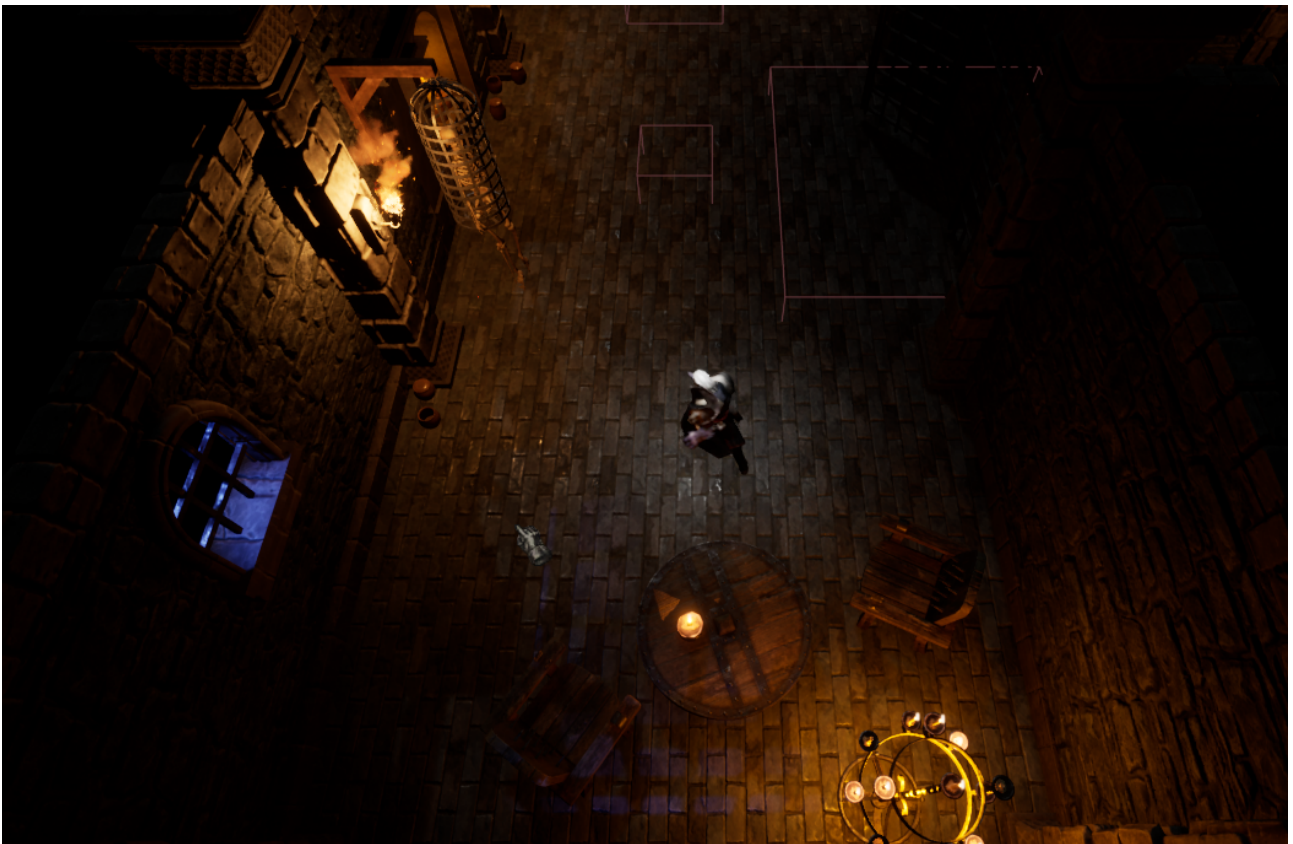
A.1 Diagrames de casos d'ús client web

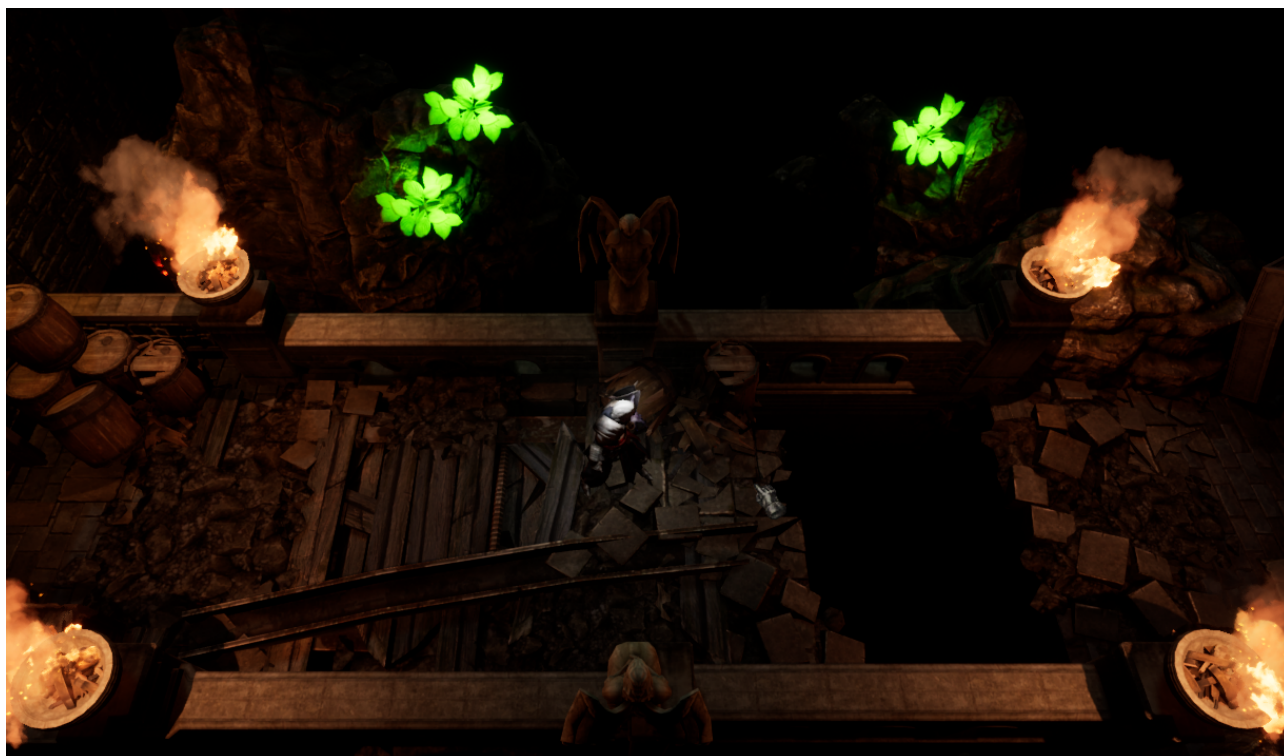


A.2 Diagrames de casos d'ús client unreal

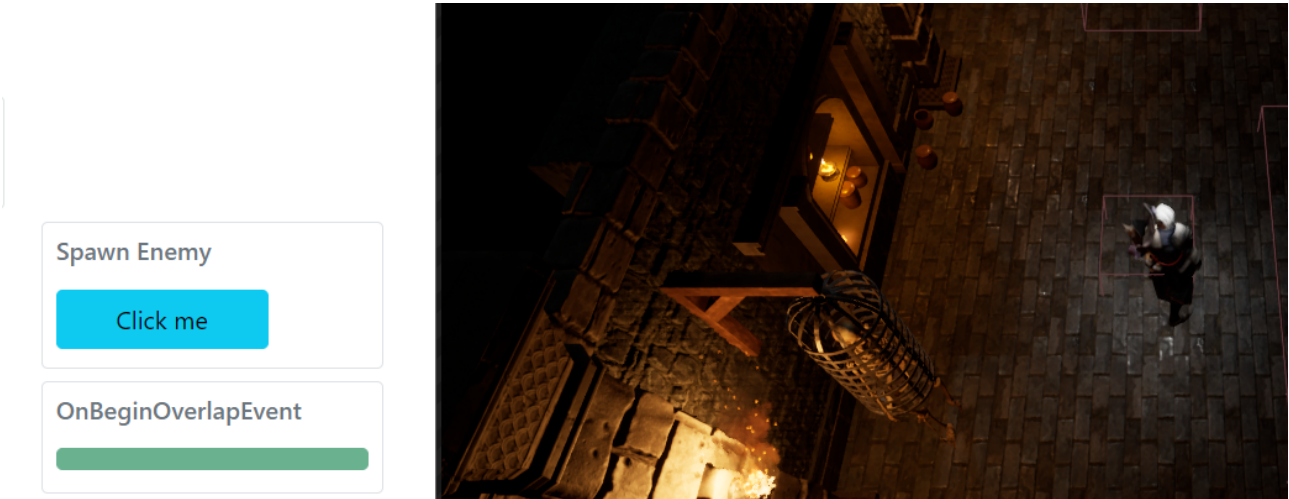


A.3 Escena inicial





A.4 Esdeveniment OnBeginOverlap



A.5 Esdeveniment Spawn Actor



A.6 Esdeveniment Spawn Item

Spawn Item

- ☒ Health Potion
☐ Boooomb!

Click me



Spawn Item

- ☐ Health Potion
☒ Boooomb!

Click me



A.7 Floating platform

Floating platform

☒ Enable/Disable platform

